Decision Trees

February 24, 2017

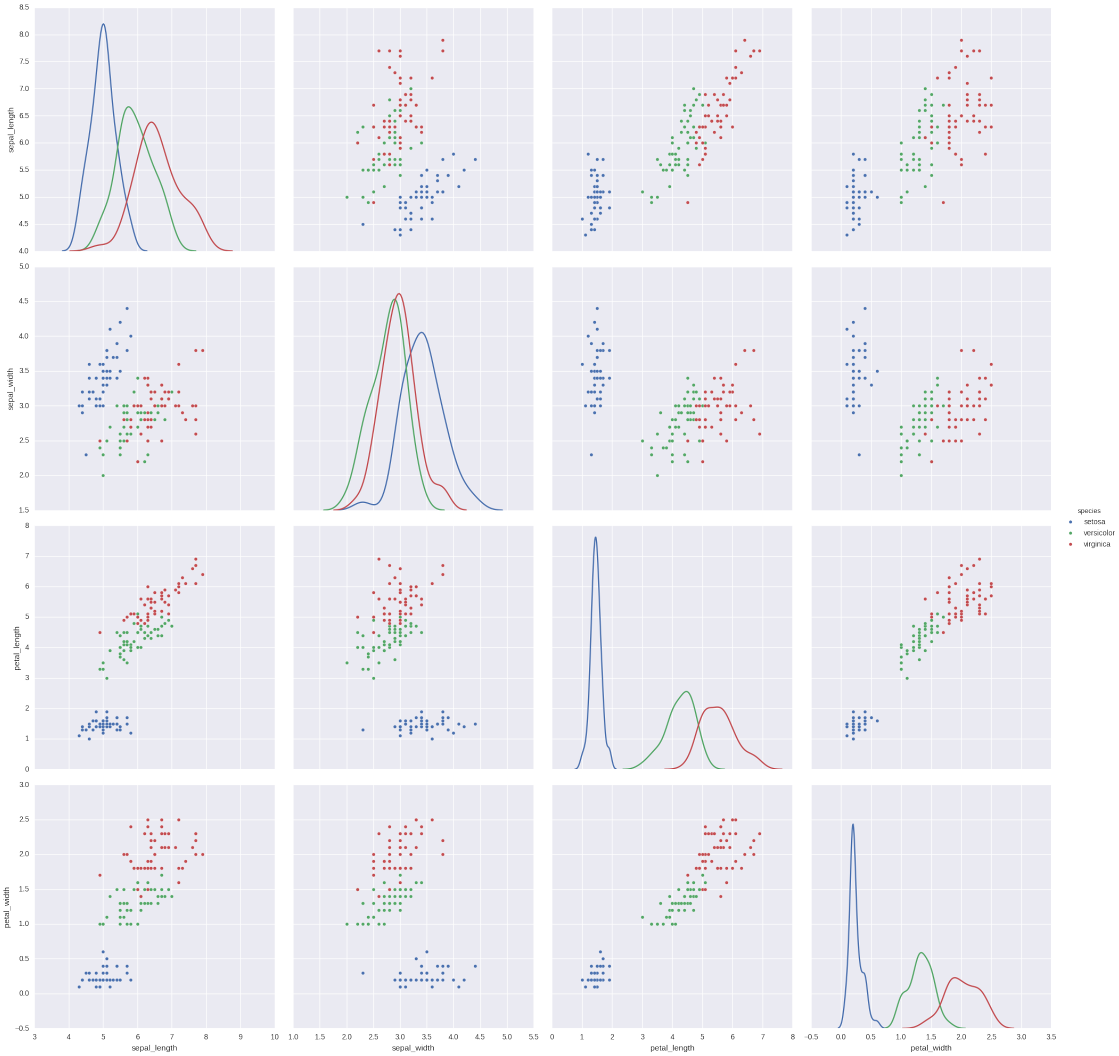
```
In [37]: import seaborn as sns

         % matplotlib inline
```

```
In [38]: iris     = sns.load_dataset('iris')
         nsamples = 5
         iris.sample(nsamples)
```

```
Out[38]:      sepal_length  sepal_width  petal_length  petal_width     species
         69            5.6          2.5           3.9          1.1  versicolor
         98            5.1          2.5           3.0          1.1  versicolor
         10            5.4          3.7           1.5          0.2      setosa
         71            6.1          2.8           4.0          1.3  versicolor
         133           6.3          2.8           5.1          1.5   virginica
```

```
In [39]: sns.pairplot(iris, hue = 'species', diag_kind = 'kde', size = 5)
```

```
Out[39]: <seaborn.axisgrid.PairGrid at 0x7fccac692668>
```



```
In [40]: from sklearn.preprocessing import LabelEncoder
```

```
In [41]: encoder            = LabelEncoder()
         iris['species'] = encoder.fit_transform(iris['species'])
```

```
In [42]: iris.sample(nsamples)
```

```
Out[42]:      sepal_length  sepal_width  petal_length  petal_width  species
         51            6.4          3.2           4.5          1.5        1
         30            4.8          3.1           1.6          0.2        0
         8             4.4          2.9           1.4          0.2        0
         0             5.1          3.5           1.4          0.2        0
         139           6.9          3.1           5.4          2.1        2
```

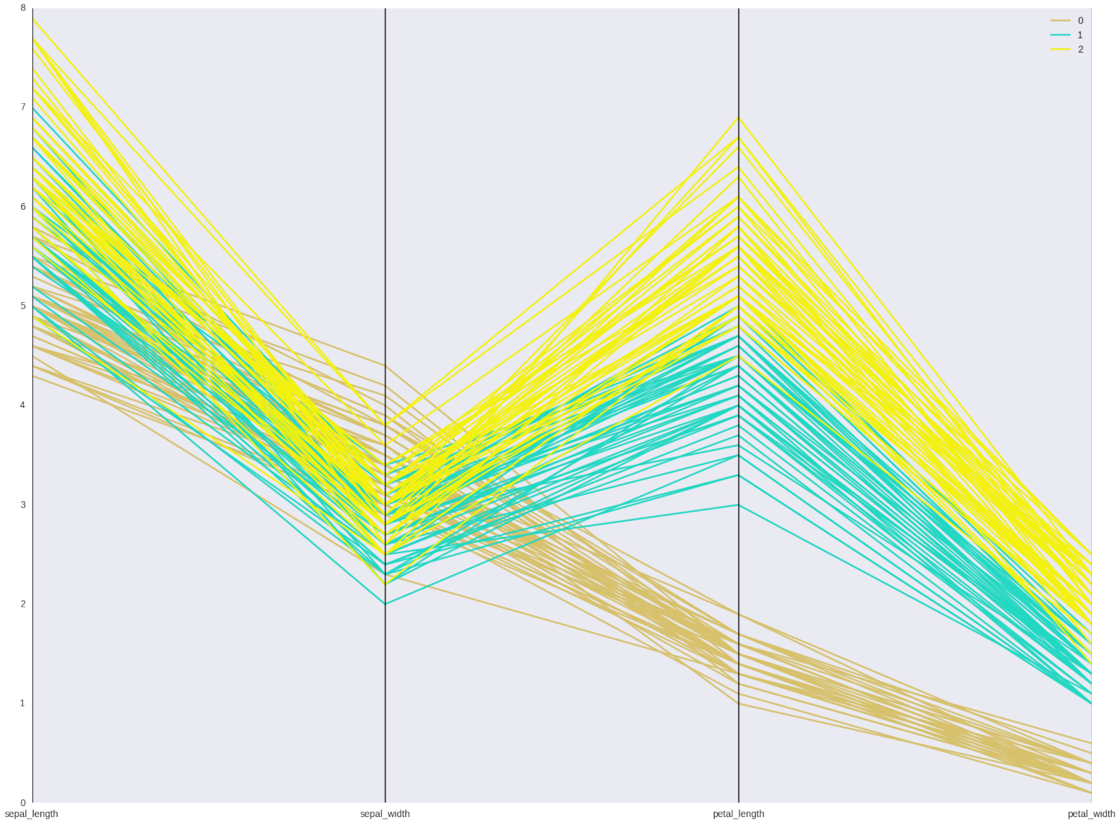
```
In [43]: from pandas.tools.plotting import parallel_coordinates, andrews_curves
```

In [44]:

```
import matplotlib.pyplot as plt

width, height = 20, 15
size          = width, height

plt.figure(figsize = size)

parallel_coordinates(iris, 'species')
```

Out[44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fccacd720f0>
```

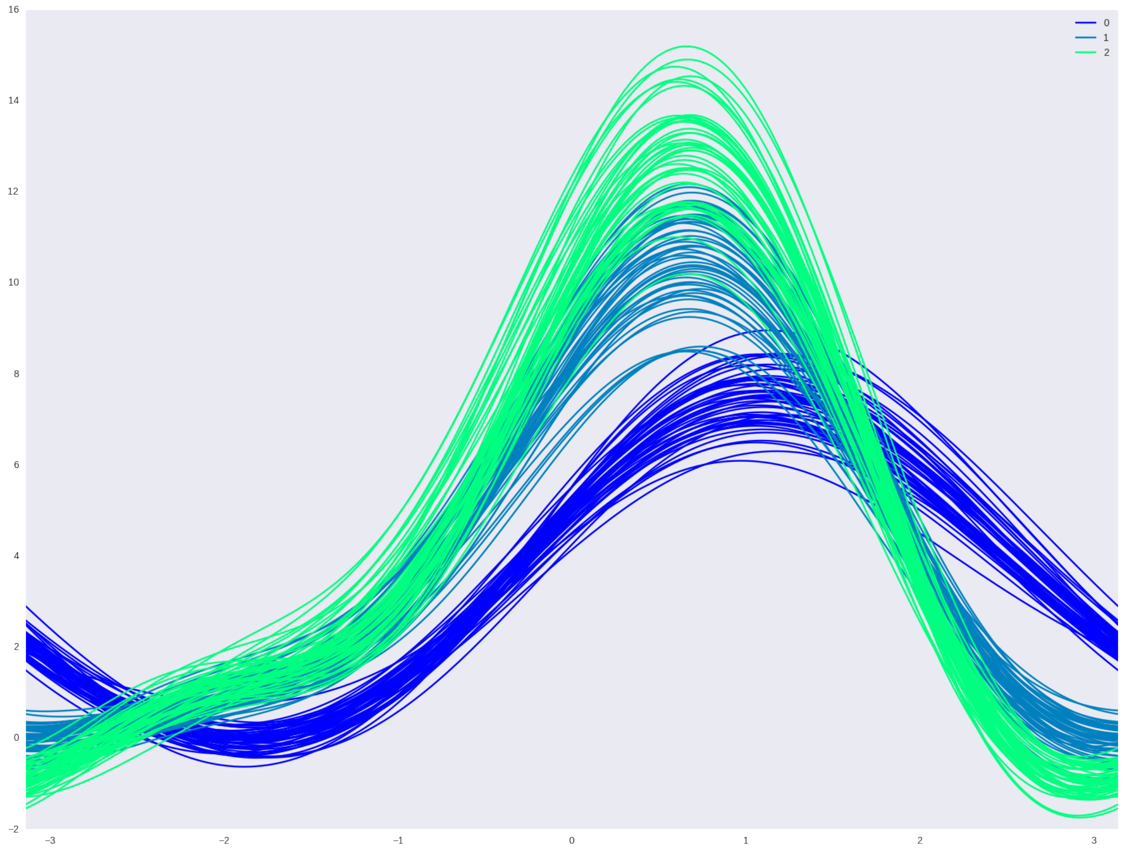


In [45]:

```
plt.figure(figsize = size)

andrews_curves(iris, 'species', colormap = 'winter')
```

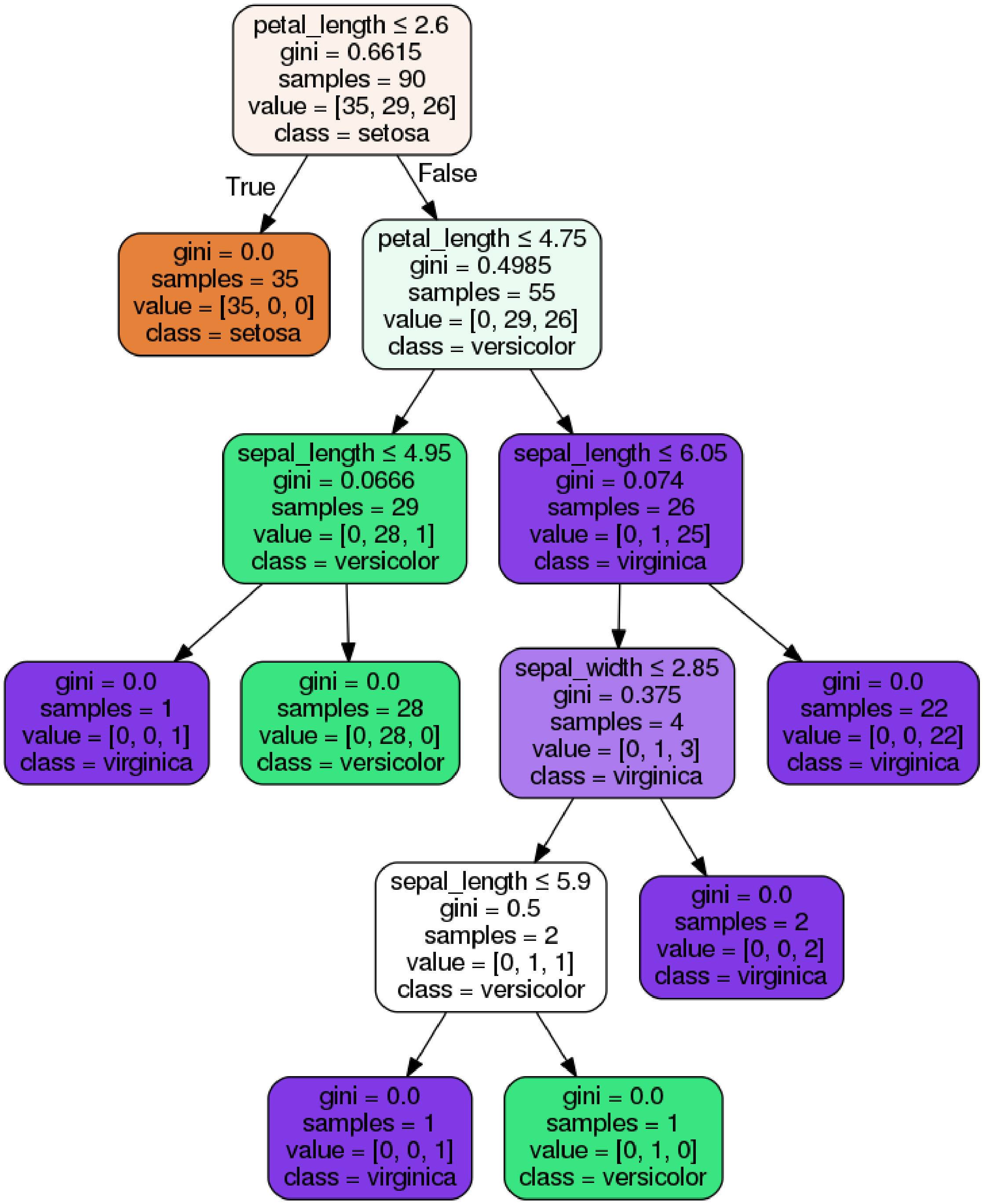
Out[45]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcc64118630>
```



```
In [46]: norm = (iris - iris.min()) / (iris.max() - iris.min())
         norm.sample(5)

Out[46]:      sepal_length  sepal_width  petal_length  petal_width  species
         6        0.083333     0.583333      0.067797     0.083333      0.0
         137      0.583333     0.458333      0.762712     0.708333      1.0
         116      0.611111     0.416667      0.762712     0.708333      1.0
         47       0.083333     0.500000      0.067797     0.041667      0.0
         104      0.611111     0.416667      0.813559     0.875000      1.0

In [47]: from sklearn.model_selection import train_test_split

In [48]: train, test = train_test_split(iris, train_size = 0.60)

In [49]: len(train), len(test)

Out[49]: (90, 60)

In [50]: from sklearn.tree import DecisionTreeClassifier

In [51]: features = iris.columns[:-2]
         target   = iris.columns[ -1]
```

```
In [52]: model = DecisionTreeClassifier()
         model = model.fit(train[features], train[target])

In [53]: from sklearn.metrics import accuracy_score

In [54]: predicted = model.predict(test[features])

In [55]: print('Accuracy Score:', accuracy_score(test[target], predicted) * 100)

Accuracy Score: 88.3333333333


In [56]: from sklearn.tree import export_graphviz

In [57]: class_names = list(set(encoder.inverse_transform(iris[target])))
         ddata = export_graphviz(model,
                                 out_file           = None,
                                 feature_names      = features,
                                 class_names        = class_names,
                                 filled             = True,
                                 rounded            = True,
                                 special_characters = True)

In [58]: import pydotplus

In [59]: graph = pydotplus.graph_from_dot_data(ddata)

In [60]: png   = graph.create_png()

In [61]: from IPython.display import Image

In [62]: Image(png)

Out[62]:
```



```
In [63]: model = DecisionTreeClassifier(criterion = 'entropy')
         model = model.fit(train[features], train[target])
```

```
In [64]: predicted = model.predict(test[features])
```

```
In [65]: print('Accuracy Score:', accuracy_score(test[target], predicted) * 100)
```

```
Accuracy Score: 88.3333333333
```

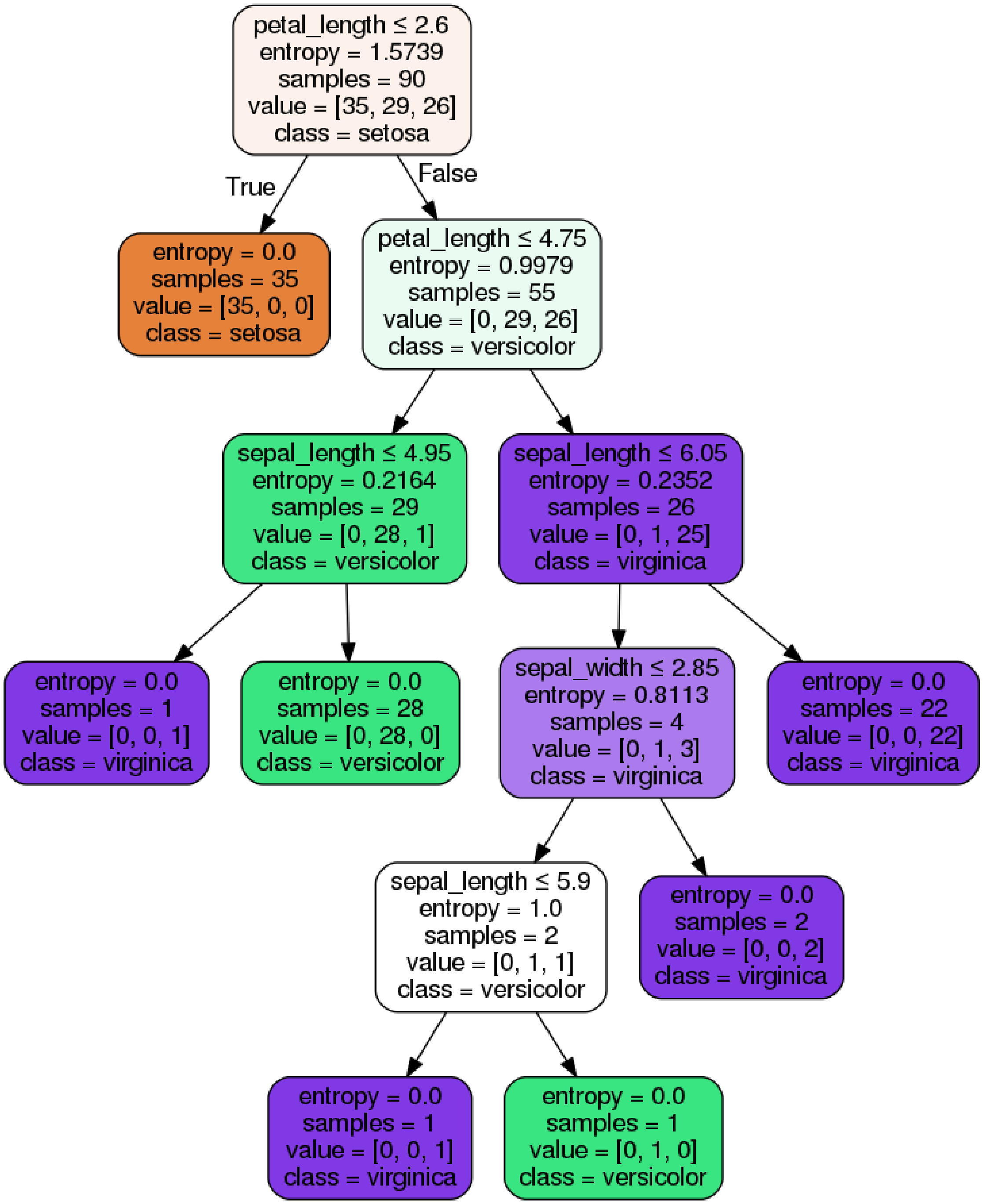
```
In [66]: ddata = export_graphviz(model,
                                out_file           = None,
                                feature_names      = features,
                                class_names        = class_names,
                                filled             = True,
                                rounded            = True,
                                special_characters = True)
```

```
In [67]: graph = pydotplus.graph_from_dot_data(ddata)
```

```
In [68]: png   = graph.create_png()
```

```
In [69]: Image(png)
```

```
Out[69]:
```


petal_length ≤ 2.6
entropy = 1.5739
samples = 90
value = [35, 29, 26]
class = setosa
True
False
entropy = 0.0
samples = 35
value = [35, 0, 0]
class = setosa
petal_length ≤ 4.75
entropy = 0.9979
samples = 55
value = [0, 29, 26]
class = versicolor
sepal_length ≤ 4.95
entropy = 0.2164
samples = 29
value = [0, 28, 1]
class = versicolor
sepal_length ≤ 6.05
entropy = 0.2352
samples = 26
value = [0, 1, 25]
class = virginica
entropy = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
entropy = 0.0
samples = 28
value = [0, 28, 0]
class = versicolor
sepal_width ≤ 2.85
entropy = 0.8113
samples = 4
value = [0, 1, 3]
class = virginica
entropy = 0.0
samples = 22
value = [0, 0, 22]
class = virginica
sepal_length ≤ 5.9
entropy = 1.0
samples = 2
value = [0, 1, 1]
class = versicolor
entropy = 0.0
samples = 2
value = [0, 0, 2]
class = virginica
entropy = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
entropy = 0.0
samples = 1
value = [0, 1, 0]
class = versicolor

```
In [ ]:
```